IN THE CLAIMS

The Claims as they currently stand are presented below.

1. (Currently amended) A programmable, architecturally-systolic, Reed-Solomon BCH error correction decoder for decoding a predetermined number of Reed-Solomon and BCH codes, said decoder comprising:

a translator circuit for receiving one of the predetermined number of Reed-Solomon and BCH codes that each have predetermined external Galois-field representations and for translating the external Galois-field representation of the received code into an internal ~~Galois-field representation comprising a~~ quadratic-subfield Galois-field representation of the code;

a syndrome computation module for calculating syndromes comprising intermediate values required to find error locations and values;

a Berlekamp-Massey computation module that implements a Berlekamp-Massey algorithm that converts the syndromes to intermediate results comprising lambda and omega polynomials;

a Chien-Forney module comprising modified Chien-search and Forney algorithms to calculate actual error locations and error values that correspond to an error-corrected code; and

an inverse translator circuit for translating the internal quadratic-subfield Galois-field representation of the error-corrected code into the external Galois-field representation.

2. (Canceled)

3. (Currently amended) The decoder recited in Claim 1 wherein the Berlekamp-Massey computation module carries out repeated dot product calculations between vectors having up to T+1 components using Galois-field arithmetic, where T is the error correcting capability of the code.

4. (Currently amended) The decoder recited in Claim 3 wherein the Berlekamp-Massey computation module comprises quadratic-subfield modular multipliers that carry out dot product calculations in a single operation.

5. (Original) The decoder recited in Claim 1 wherein the Berlekamp-Massey computation module and the Chien-Forney module each include a quadratic-subfield-power integrated divider that carries out Galois-field division in a quadratic-subfield representation.

6. (Original) The decoder recited in Claim 1 wherein the Chien-Forney module comprises an offset-adjustment-free Forney module that carries out Forney's algorithm without calculating a formal derivative of the lambda polynomial and without calculating an offset-adjustment factor for Reed-Solomon codes with offsets in the code-generator polynomial.

7. (Original) The decoder recited in Claim 1 wherein the clocks controlling the syndrome computation module, the Berlekamp-Massey computation module, and the Chien-Forney module are separate and free-running clocks requiring no fixed phase relationship, to allow maximum speed and flexibility for the clocks of each module.

8. (Original) The decoder recited in Claim 1 wherein configuration information travels systolically with the data from the syndrome module to the Berlekamp-Massey module and from the Berlekamp-Massey module to the Chien-Forney module, providing for switching among different codes and among codes of different degrees of shortening.

9. (Previously presented) The decoder recited in Claim 1 wherein two simultaneous BCH data blocks are processed at once.

10. (Original) The decoder recited in Claim 1 wherein internal registers and computation circuitry are shared among different code types, binary BCH and non-binary Reed-Solomon, thereby reducing total gate count.

11. (Original) The decoder recited in Claim 1 wherein alterations solely in exclusive-OR trees of the translator and inverse translator circuits enable the decoder to decode Reed-Solomon codes using any Galois-field representation linearly related to standard representations, including representations generated by a field-generator polynomial and standard subfield representations.

12. (Original) The decoder recited in Claim 1 wherein alterations solely in exclusive-OR trees of the syndrome module and the Chien-Forney module enable the decoder to decode Reed-Solomon codes using code-generator polynomials having any offset and skip values, including standard code-generator polynomials.

13. (Original) The decoder recited in Claim 1 wherein logic checks in the Berlekamp-Massey module on the length of the lambda polynomial and in the Chien-Forney module on the number of errors detected are sufficient to detect all undetectable error patterns that are mathematically possible to detect.

14. (Previously presented) A method for decoding a predetermined number of Reed-Solomon and BCH codes comprising the steps of:

translating one of a predetermined number of Reed-Solomon and BCH codes that each have predetermined external Galois-field representations into an internal quadratic subfield Galois-field representation;

calculating syndromes comprising intermediate values required to find error locations and values;

converting the syndromes to intermediate results comprising lambda and omega polynomials using a Berlekamp-Massey algorithm;

calculating actual error locations and error values that correspond to an error-corrected code using Chien-search and Forney algorithms; and

translating the internal quadratic subfield Galois-field representation of the error-corrected code into the external Galois-field representation.

15. (Canceled)

16. (Previously presented) The method recited in Claim 14 wherein the step of converting the syndromes to intermediate results comprises the steps of performing repeated dot product calculations between vectors having up to T+1 components using Galois-field arithmetic, where T is the error correcting capability of the code.

17. (Original) The method recited in Claim 14 wherein alterations solely in exclusive-OR trees enable decoding of Reed-Solomon codes using any Galois-field representation linearly related to standard representations, including representations generated by a field-generator polynomial and standard subfield representations.

18. (Original) The method recited in Claim 14 wherein alterations solely in exclusive-OR trees enable decoding of Reed-Solomon codes using code-generator polynomials having any offset and skip values, including standard code-generator polynomials.

19. (Original) The method recited in Claim 14 wherein logic checks on the length of the lambda polynomial and on the number of errors detected are sufficient to detect all undetectable error patterns that are mathematically possible to detect.

20. (Currently amended) A programmable, architecturally-systolic, Reed-Solomon BCH error correction decoder for decoding a predetermined number of Reed-Solomon and BCH codes, said decoder comprising:

a translator circuit for receiving one of the predetermined number of Reed-Solomon and BCH codes that each have predetermined external Galois-field representations and for translating the external Galois-field representation of the received code into an internal ~~Galois-field representation comprising a~~ quadratic-subfield Galois-field representation of the code;

a syndrome computation module for calculating syndromes comprising intermediate values required to find error locations and values;

a Berlekamp-Massey computation module comprising quadratic-subfield multipliers that implements a Berlekamp-Massey algorithm that converts the syndromes to intermediate results comprising lambda and omega polynomials;

a Chien-Forney module comprising modified Chien-search and Forney algorithms to calculate actual error locations and error values that correspond to an error-corrected code; and

an inverse translator circuit for translating the internal quadratic-subfield Galois-field representation of the error-corrected code into the external Galois-field representation.